

Docket No.: 56937-111

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Customer Number: 20277
	:	
Katsuya SHINOHARA	:	Confirmation Number:
	:	
Serial No.:	:	Group Art Unit:
	:	
Filed: April 01, 2004	:	Examiner: Unknown
	:	
For: SIMULATOR APPARATUS AND RELATED TECHNOLOGY	:	

**CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Mail Stop CPD
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 35 U.S.C. 119, Applicant hereby claims the priority of:

Japanese Patent Application No. 2003-104907, filed April 9, 2003

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY


Michael E. Fogarty
Registration No. 36,139

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 MEF:tlb
Facsimile: (202) 756-8087
Date: April 1, 2004

56937-111
SHINOHARA
April 1, 2004

日 本 国 特 許 庁
JAPAN PATENT OFFICE

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 4 月 9 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 1 0 4 9 0 7
Application Number:

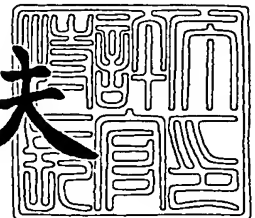
[ST. 10/C]: [J P 2 0 0 3 - 1 0 4 9 0 7]

出 願 人 松下電器産業株式会社
Applicant(s):

2 0 0 3 年 1 1 月 2 1 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 0 9 6 6 5 8

【書類名】 特許願

【整理番号】 5037540160

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 17/50
F24D 13/02

【発明者】

【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社
会社内

【氏名】 篠原 克哉

【特許出願人】

【識別番号】 000005821

【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100086737

【弁理士】

【氏名又は名称】 岡田 和秀

【電話番号】 06-6376-0857

【手数料の表示】

【予納台帳番号】 007401

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9305280

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 シミュレータ装置、シミュレーション方法および性能解析方法

【特許請求の範囲】

【請求項 1】 被シミュレーション対象システムを構成する CPU、前記 CPU や周辺のハードウェアのブロックを接続するバスおよび前記ハードウェアをシミュレートするシミュレータ装置であって、前記 CPU のモデル、バス、および前記バスに接続される前記ハードウェアのモデルとのインターフェースを実装し、前記インターフェースを用いて、前記被シミュレーション対象システムの性能／機能検証を実行することを特徴とするシミュレータ装置。

【請求項 2】 前記各モデルのインターフェースが、ソフトウェア開発検証用システムシミュレータで利用できるインターフェースを備えていることを特徴とする請求項 1 に記載のシミュレータ装置。

【請求項 3】 前記各モデルのインターフェースが、ハードウェア検証用システムシミュレータで利用できるインターフェースを備えていることを特徴とする請求項 1 に記載のシミュレータ装置。

【請求項 4】 前記各モデルのインターフェースが、システム検証用システムシミュレータで利用できるインターフェースを備えていることを特徴とする請求項 1 に記載のシミュレータ装置。

【請求項 5】 前記各モデルのインターフェースが、デバッグで利用できるインターフェースを備えていることを特徴とする請求項 1 に記載のシミュレータ装置。

【請求項 6】 請求項 5 に記載のインターフェースを用いて、デバッグすることを特徴とするシミュレーション方法。

【請求項 7】 システムの 1 クロックサイクル数分のシミュレーションを実行するためのインターフェースを備えていることを特徴とする請求項 1 に記載のシミュレータ装置。

【請求項 8】 請求項 7 に記載の前記シミュレータ装置を用いてシステムの 1 クロックサイクル数分のシミュレーションを実行することを特徴とするシミュレーション方法。

【請求項 9】 システムのシミュレーション時間分のシミュレーションを実行するためのインターフェースを備えていることを特徴とする請求項 1 に記載のシミュレータ装置。

【請求項 10】 請求項 9 に記載の前記インターフェースを用いて、システムのシミュレーション時間分のシミュレーションを実行することを特徴とするシミュレーション方法。

【請求項 11】 前記各モデルのインターフェースが、性能解析で使用できる拡張用インターフェースを備えていることを特徴とする請求項 1 に記載のシミュレータ装置。

【請求項 12】 請求項 11 に記載の前記インターフェースを用いて、性能解析することを特徴とするシミュレーション方法。

【請求項 13】 請求項 11 に記載の前記インターフェースを用いて生成された情報を基に、性能解析することを特徴とする性能解析方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、大規模集積回路（LSI）設計、特にシステムオンチップ化に対応するための設計に関するものであり、設計上流におけるハードウェア／ソフトウェア／システムの検証、開発、評価を行うためのシミュレータ装置、シミュレーション方法および性能解析方法に関するものである。

【0002】

【従来の技術】

従来から、電子機器は、例えばプロセッサ、メモリ、バス等の種類ごとの LSI を、個別に半導体チップ上に形成した後、各チップをプリント配線基板などの母基板上に実装することによって製造されてきた。

【0003】

ところが、近年、これに用いられる集積回路装置に対し、1チップ化、小型化、軽量化、省電力化および低コスト化の要求が高まっている。このような傾向は、特にデジタル情報家電分野において、より顕著にみられる。そして、これに応

じて、半導体メーカーはその研究開発の重心をメモリからシステム L S I へと移行してきている。

【 0 0 0 4 】

システムはソフトウェアとハードウェアから構成される。ハードウェアとソフトウェアに分けられた後、ハードウェアとソフトウェアを逐次的に開発していた。このようにハードウェアとソフトウェアを逐次的に開発すると、開発期間の増大につながるため、ハードウェアとソフトウェアを同時に並行開発するハードウェア／ソフトウェア協調設計という手法がとられつつある。ハードウェアの検証用にシステムシミュレータ、エミュレータを用い、ソフトウェアの開発／検証用にも別途、システムシミュレータや評価ボードを用いて、それぞれの開発／検証を行うのである。

【 0 0 0 5 】

従来のシミュレーション技術は、異なる用途向けの、異なる抽象度で記述されたモデルを想定している（例えば、特許文献 1 参照）。

【 0 0 0 6 】

また、互いに異なる抽象度で書かれたモデルを統一するシミュレーション方法も提案されている（例えば、特許文献 2 参照）。

【 0 0 0 7 】

【特許文献 1】

特開 2 0 0 2 - 1 5 7 2 9 1 号公報（第 6 - 7 頁、第 2 図）（半導体集積回路装置の設計方法）

【特許文献 2】

特願 2 0 0 2 - 1 6 2 0 4 8 号（シミュレータ装置並びにシミュレーションモデル生成プログラム）

【 0 0 0 8 】

【発明が解決しようとする課題】

しかしながら、ハードウェア／ソフトウェア協調設計を行うためには、そのための設計環境が必要となる。すなわち、システムシミュレータが必要となる。ハードウェアとソフトウェアの分割に使用されるシステム検証用シミュレータと、

ハードウェアの検証に使用されるハードウェア検証用システムシミュレータと、ソフトウェアの開発検証用に用いられるソフトウェア開発検証用シミュレータと、それぞれの用途向けに、プロセッサ、バス、ハードウェアモデルを用意するのが現状である。

【0009】

しかし、システムLSIの複雑化・大規模化を鑑みると、それらのモデルを用意するための工数も同様に増加の一途をたどっている。結果、ハードウェア／ソフトウェア協調設計を行うことで得られる恩恵と同程度かそれ以上の工数が発生し、ハードウェア／ソフトウェア協調設計の利点が失われてしまう。

【0010】

本発明は上記問題点に鑑み、それぞれのシステムシミュレータで利用できる共通のモデルを用いたシミュレータ装置、シミュレーション方法、性能解析方法を提供することを目的とする。

【0011】

【課題を解決するための手段】

以上の目的を達成するための本発明のシステムシミュレータ装置およびシミュレーション方法は、大規模なシステムLSIの1チップシミュレーションで用いるブロックについて、異なるシミュレータの制御部からブロックを呼び出し可能なインターフェースを備えたものである。また、デバッグ用のインターフェースを実装することで、各用途のための検証解析機能を実現するものである。以下、順次説明する。

【0012】

本発明によるシミュレータ装置は、被シミュレーション対象システムを構成するCPU、前記CPUや周辺のハードウェアのブロックを接続するバスおよび前記ハードウェアをシミュレートするシミュレータ装置であって、前記CPUのモデル、バス、および前記バスに接続される前記ハードウェアのモデルとのインターフェースを実装し、前記インターフェースを用いて、前記被シミュレーション対象システムの性能／機能検証を実行することを特徴とする。

【0013】

上記構成において、前記各モデルのインターフェースとしては、ソフトウェア開発検証用システムシミュレータで利用できるインターフェースや、ハードウェア検証用システムシミュレータで利用できるインターフェースや、システム検証用システムシミュレータで利用できるインターフェースなどがある。

【0014】

この構成によれば、CPUのモデルとハードウェアのモデルを構成するとともに、両モデル間にインタフェースを介在させていて、ソフトウェア検証向け、システム検証向け、ハードウェア検証向けなど各種の検証向けにモデルを共通化することができる。すなわち、大規模なシステムLSIのハードウェア検証用システムシミュレータ、ソフトウェア開発検証用シミュレータ、システム検証用システムシミュレータをはじめとする各種目的用システムシミュレータにおいて、共通のモデルが利用可能になるため、異なるシミュレータ間でモデルの共有化が図られ、統合シミュレーション制御方法によって、適切な精度のシミュレーションを同一のモデルを用いて実行できる。したがって、それぞれの目的ごとにモデルを用意する必要がなくなる。

【0015】

また、上記シミュレータ装置の構成において、前記各モデルのインターフェースとしては、デバッグで利用できるインターフェースもある。これに対応するシミュレーション方法は、そのデバッグ用のインターフェースを用いてデバッグすることである。例えば、ブロックの機能としてデータ転送の開始、終了といった処理が行われた場合にブレークを行うようにするデバッグ用のインターフェースを備えることで、各用途向けモデルで共通して利用でき、デバッグ性が向上する。

【0016】

さらに、上記シミュレータ装置の構成において、前記各モデルのインターフェースとしては、システムの1クロックサイクル数分のシミュレーションを実行するためのインターフェースもある。これに対応するシミュレーション方法は、そのシステムの1クロックサイクル数分のシミュレーションを実行するためのインターフェースを用いて、システムの1クロックサイクル数分のシミュレーション

を実行することである。各モデルを駆動するタイミングがクロックレベルであるときに有効である。

【0017】

また、上記シミュレータ装置の構成において、前記各モデルのインターフェースとしては、システムのシミュレーション時間分のシミュレーションを実行するためのインターフェースもある。これに対応するシミュレーション方法は、そのシステムのシミュレーション時間分のシミュレーションを実行するためのインターフェースを用いて、システムのシミュレーション時間分のシミュレーションを実行することである。各モデルで使用するクロックサイクルが異なる場合に有効である。すなわち、異なるクロックソースを用いるモデル混在のシミュレーション環境が実現できる。

【0018】

また、上記シミュレータ装置の構成において、前記各モデルのインターフェースとしては、性能解析で利用できる拡張用インターフェースもある。これに対応するシミュレーション方法または性能解析方法は、その解析で利用できる拡張用インターフェースを用いて解析することである。性能解析の結果を基にシステムをチューニングして、最適なシステムを設計することが可能となる。

【0019】

【発明の実施の形態】

以下、本発明の一実施形態について、図面を参照しながら説明する。

【0020】

図1は本発明のシミュレータ装置の構成を示すブロック図である。図中、検証用シミュレータ装置1は、被シミュレーション対象システムのシミュレータ装置である。被シミュレーション対象システムは、ブロック31、ブロック32、ブロック33から構成された場合の例である。ブロック31のモデルは、ソフトウェア検証シミュレータ用I/F3と、ハードウェア検証用シミュレータ用I/F4と、システム検証シミュレータ用I/F5と、デバッグ用I/F6と、拡張用I/F7とを備え、ブロックの機能モデル8からなるシミュレータモデル9として、検証用シミュレータ装置1を構成する。

【0021】

同様に、ブロック 32 のモデルは、ソフトウェア検証シミュレータ用 I/F 10 と、ハードウェア検証用シミュレータ用 I/F 11 と、システム検証シミュレータ用 I/F 12 と、デバッグ用 I/F 13 と、拡張用 I/F 14 とを備え、ブロックの機能モデル 15 からなるシミュレータモデル 16 として、検証用シミュレータ装置 1 を構成する。

【0022】

さらに、ブロック 33 のモデルは、ソフトウェア検証シミュレータ用 I/F 17 と、ハードウェア検証用シミュレータ用 I/F 18 と、システム検証シミュレータ用 I/F 19 と、デバッグ用 I/F 20 と、拡張用 I/F 21 とを備え、ブロックの機能モデル 22 からなるシミュレータモデル 23 として、検証用シミュレータ装置 1 を構成する。

【0023】

検証用シミュレータ装置 1 がハードウェア検証用であれば、モデル 9、モデル 16、モデル 23 のそれぞれのハードウェア検証用シミュレータ用 I/F 4、ハードウェア検証用シミュレータ用 I/F 11、ハードウェア検証用シミュレータ用 I/F 18 を用いてハードウェア検証用のシミュレータ制御手段 2 がシミュレーションの制御を行う。

【0024】

また、検証用シミュレータ装置 1 がソフトウェア検証用であれば、モデル 9、モデル 16、モデル 23 のそれぞれのソフトウェア検証シミュレータ用 I/F 3、ソフトウェア検証シミュレータ用 I/F 10、ソフトウェア検証シミュレータ用 I/F 17 を用いてソフトウェア検証用のシミュレータ制御手段 2 がシミュレーションの制御を行う。

【0025】

さらに、検証用シミュレータ装置 1 がシステム検証用であれば、モデル 9、モデル 16、モデル 23 のそれぞれのシステム検証シミュレータ用 I/F 5、システム検証シミュレータ用 I/F 12、システム検証シミュレータ用 I/F 19 を用いてシステム検証用のシミュレータ制御手段 2 がシミュレーションの制御を行

う。

【0026】

それぞれの目的における検証時、内部の状態、例えばブロックのレジスタの値を知る必要がある。そのような場合には、デバッグ用 I/F 6、デバッグ用 I/F 13、デバッグ用 I/F 20 を用いて、内部の状態を取り出し、デバッグに利用する。レジスタ以外の信号の値を含め、デバッグ目的で必要となる情報は、すべてこの I/F により実装される。

【0027】

また、システムシミュレータにおいては、システムの振る舞いを基に、性能解析する必要が発生する。複数のバスマスタからなるマルチバスマスタシステムにおけるバスの負荷や、マスタが CPU であった際の CPU 利用率、メモリの転送レートや利用率が、そのような解析で必要となるデータである。各モデルは、バスの利用率を計算するために必要な、バス要求時やバス利用許可時、バス利用放棄時などを、このインターフェースを通して、シミュレータ制御手段をはじめとする解析系にデータを渡す。

【0028】

簡単な解析系としては、インターフェースがそれらのデータを、電子データとしてファイルに保存し、それを利用して解析するものが挙げられる。

【0029】

ブロック 31、ブロック 32、ブロック 33 のレジスタの内容を記憶することをはじめとする機能は、機能モデル 8、機能モデル 15、機能モデル 22 でモデル化されるものとし、前述 I/F を用いて、機能を駆動する。レジスタへのアクセスや、デバッグのアクセスは、すべてこの機能モデルで実装される。このようなシミュレータ装置の実装をとることで、それぞれの目的ごとにモデルを用意する必要がなくなる。

【0030】

図 2 は、被シミュレーション対象システムの LSI であるシステム LSI 30 の構成図である。システム LSI 30 は、ブロック 31、ブロック 32、ブロック 33、ブロック 34、ブロック 35、ブロック 36、ブロック 37 とバスから

構成される。

【0031】

図3は、被シミュレーション対象システムのLSIであるシステムLSI30のハードウェア検証用シミュレータ60の1実現例である。ハードウェア検証用シミュレータ60は、システムLSI30の構成要素ブロック31、ブロック32、ブロック33、ブロック34、ブロック35、ブロック36、ブロック37とバスのそれぞれに対応して、検証用のモデル61、モデル62、モデル63、モデル64、モデル65、モデル66、モデル67、およびバスモデル68とバスモデル69から構成されている。

【0032】

検証対象ハードウェアがブロック31であるとし、検証対象ハードウェアのモデルをモデル61とする。ハードウェアのモデルは、ハードウェア記述言語（HDL：Hardware Description Language）と呼ばれる記述言語、もしくは高位合成で設計されたモデルを含む。この目的のためには、検証対象ハードウェアブロックに影響を与える各モデルがクロックサイクルか、もしくはRTL（Register Transfer Level：レジスタ転送レベル）と呼ばれるレベルで検証対象ハードウェアブロックにアクセスする必要がある。

【0033】

図6は、クロックレベルで必要とされる精度を示している。システムレベルの場合は、図中、時刻 t 、時刻 $t+1$ 、時刻 $t+2$ 、時刻 $t+3$ 、時刻 $t+4$ での各信号の値がピンレベルで、実際のハードウェア記述モデルと同じであると保証されていることが最低限必要となる。

【0034】

図4（a）は、被シミュレーション対象システムのLSIであるシステムLSI30のソフトウェア検証用シミュレータ90の1実現例である。ブロック31が検証対象ソフトウェアが動作するブロックとし、そのモデルを91とする。一般的には、ブロック31のISSと呼ばれる命令レベルでの精度を保証する命令セットシミュレータと、ブロック32、ブロック33、ブロック34、ブロック35、ブロック36、ブロック37のメモリアメージをはじめとする、検証対象

ソフトウェアの動作に必要な機能を実装した仮想モデル 92 から構成され、検証対象ソフトウェアがアクセスする部分は、仮想モデルとして実装される。ミドルウェアやデバイスドライバと呼ばれるハードウェアモデルの高精度を求めるソフトウェアが検証対象であると、この仮想モデルは、図 3 のハードウェア検証用シミュレータに近い精度が求められることもある。この目的で作られるシミュレータは、モデル 91 がシミュレータの制御を行い、モデル 91 の外部へアクセスが発生したときにのみ仮想モデル 92 が使用される、といったモデル化がよく行われる。図 4 (b) のような場合もある。

【0035】

図 7 は、このレベルで必要とされる精度を示している。各命令が影響を与えるブロックに対する影響が、検証対象ソフトウェアの動作するブロック 31 のモデル 91 の内部ではなく、外部に、すなわち図の例では仮想モデル 92 に及ぶことが最低限必要となる。

【0036】

図 5 は、被シミュレーション対象システムの LSI であるシステム LSI 30 のシステム検証用シミュレータ 120 の 1 実現例である。ブロック 31、ブロック 32、ブロック 33、ブロック 34、ブロック 35、ブロック 36、ブロック 37 のメモリアメージをはじめとする、検証対象システムの動作に必要な機能を実装したそれぞれのモデル、すなわち、モデル 121、モデル 122、モデル 123、モデル 124、モデル 125、モデル 126、モデル 127 とバスモデル 128、バスモデル 129 から構成され、検証対象システム上で動作するソフトウェアがアクセスする部分は、全て実装される。

【0037】

図 8 は、このレベルで必要とされる精度を示した一例である。図 6 と図 7 の間の精度を必要としている。図中にあるように、サイクル C、サイクル C₊₁、サイクル C₊₂、サイクル C₊₃、サイクル C₊₄ の各サイクルで精度を保証する図のモデルは、その一例である。また、より抽象度の高いトランザクションでの精度を保証するシミュレータも、この範疇に含まれる。

【0038】

ソフトウェア検証用シミュレータを、他の検証用シミュレータで利用するためには、まず、システムシミュレータ向けに、制御をシミュレータ制御手段2に行わせる、図9のようなソフトウェア検証シミュレータ用I/F3を用意すればよい。

【0039】

従来のソフトウェア検証用シミュレータは、検証対象ソフトウェアが動作するプロセッサのみをモデル化し、バス、メモリに対するアクセスは抽象化されていた。一例を挙げると、図10のフローチャートのような処理工程である。図10中、初期化工程100を行い、その後、ブロック31のシミュレーション処理工程101を行う。このとき、バス、メモリに対する処理も、この処理工程内で行われる。そのため、図4(b)に示すような複数のプロセッサからなるマルチプロセッサ構成では対応できない。

【0040】

そこで、図9中、シミュレータ制御手段2により呼び出し可能な関数を定義し、その関数の実行時、外部へ影響を与える場合、前記ソフトウェア検証シミュレータ用I/F3を通して外部へアクセスを行う。

【0041】

こうすることで、システムシミュレータとして利用することが可能となる。その処理工程を図11のフローチャートに示す。図11中、初期化工程105を行い、その後、シミュレーションが終わったかを工程106で調べる。シミュレーションが終わっていれば、シミュレーションを終了する。終わっていなければ、ブロック31の機能モデル8の内部の値を更新するブロック31のupdate処理工程107を行う。このとき、バス、メモリに影響を与える処理は行われない。次工程のブロック31のcommunication処理工程108にて、機能モデル8外への処理を行う。このとき、関係するブロックのインターフェースを介してアクセスを行う。

【0042】

以上、この処理工程を実装するものが、シミュレータ制御手段2である。上述ソフトウェア検証シミュレータ用I/F3の実装を行うことで、ソフトウェアシ

ミュレータは、シミュレータ装置の一部として利用可能になる。

【0043】

このモデルをシステム検証用のシステムシミュレータで利用可能にするシステム検証シミュレータ用 I/F 5 について説明する。前述したソフトウェア検証シミュレータ用 I/F 3 は、機能モデル 8 を呼び出す。ソフトウェア検証用では、命令レベルの実装であったが、サイクルレベルで実装を行っている場合を考える。

【0044】

命令レベルとサイクルレベルとの違いを図 12 に示す。モデル 31 が 3 つのパイプラインステージから構成されているとする。図にあるように、命令レベルでは、それぞれの命令が逐次的に実行される。

【0045】

サイクルレベルでは、それぞれの命令がパイプラインステージを流れていく。サイクル C では命令 1 がステージ 1 で処理され、サイクル C₊₁ では、命令 1 がステージ 2 で、命令 2 がステージ 1 でそれぞれ処理されている。

【0046】

図 13 は、命令レベルでの処理工程を示すフローチャートである。命令レベルの処理工程のため、命令 1 の処理工程 109、命令 2 の処理工程 110、命令 3 の処理工程 111 の順に処理工程が進む。

【0047】

機能モデル 8 では、図 13 の命令 1 の処理工程 109 の処理は、図 14 のフローチャートに図示する工程で処理を行う。すなわち、ステージ 1 での処理工程 1090、ステージ 2 での処理工程 1091、ステージ 3 での処理工程 1092 の順に、シミュレータ制御手段 2 が制御を行う。

【0048】

同様にして、機能モデル 8 では、図 13 の命令 2 の処理工程 110 の処理は、図 15 のフローチャートに図示する工程で処理を行う。すなわち、ステージ 1 での処理工程 1100、ステージ 2 での処理工程 1101、ステージ 3 での処理工程 1102 の順に、シミュレータ制御手段 2 が制御を行う。

【0049】

また、同様にして、機能モデル 8 では、図 13 の命令 3 の処理工程 111 の処理は、図 16 のフローチャートに図示する工程で処理を行う。すなわち、ステージ 1 での処理工程 1110、ステージ 2 での処理工程 1111、ステージ 3 での処理工程 1112 の順に、シミュレータ制御手段 2 が制御を行う。

【0050】

サイクルレベルの場合は、これら処理工程の制御方法を、図 17 のフローチャートのように行えばよい。図 17 は、前記機能モデル 8 について、図 12 にあるサイクル C、サイクル C₊₁、サイクル C₊₂、サイクル C₊₃、サイクル C₊₄ の処理に対応し、シミュレータ制御手段 2 が、システム検証用シミュレータの制御方法を行った処理工程を示すものである。

【0051】

図中、C0 がサイクル C での処理工程（命令 1、ステージ 1）を表し、C1 がサイクル C₊₁ での処理（命令 2、ステージ 1 と命令 1、ステージ 2）を表す。図中、C2 ではサイクル C₊₂ の処理（命令 3、ステージ 1 と命令 2、ステージ 2 と命令 1、ステージ 3）を表し、C3 は、サイクル C₊₃ における処理工程（命令 3、ステージ 2 と命令 2、ステージ 3）を表し、C4 は、サイクル C₊₄ における処理工程（命令 3、ステージ 3）を表している。

【0052】

例えば処理工程 C2 にて、前記命令 3 のステージ 1 での処理工程 1110、命令 2 のステージ 2 での処理工程 1101、命令 1 のステージ 3 での処理工程 1092 を行っている一例である。ステージ間の依存関係によって、処理工程 1110、1101、1092 はどの順序で実行してもよく、逐次的にではなく、並列実行した場合にも、本発明の効果が得られることから、本発明の範疇に含まれる。

【0053】

以上のように、シミュレータ制御手段 2 が、ブロック 31 の機能モデル 8 との、システム検証シミュレータ用 I/F を用いた制御を行うことで、本発明で得られる効果が得られる。

【0054】

次に、ブロック 31 の機能モデル 8 をハードウェア検証用に用いるためのハードウェア検証用シミュレータ用 I/F 4 について説明する。

【0055】

ハードウェア検証用では、図 6 にある精度でのインターフェースが必要となる。最低限、クロックの立ち上がりと立ち下がりや、信号の値が変化したというイベントの受け渡しを、正しいタイミングで行うことを、このインターフェースでは実現する。上述したインターフェースの実装方法を用いることで、図 8 の精度を実現するシミュレータ制御手段 2 により、図 8 の精度を実現したシミュレータ装置が実現できる。

【0056】

次に、この図 8 のレベルと、ハードウェア検証用のシミュレータが必要とするレベルの実装方法を説明する。

【0057】

図 17 での実装方法では、命令 1 のステージ 1 での処理工程 1090 は、図 11 にある update と communicate の実装を行っている。これら実装されたインターフェースは、シミュレータ制御手段 2 により、例えば、クロックサイクル毎に各モデルが呼び出され、機能モデル内部の状態を変更する関数 update と、外部への通信を行うフェーズである communicate を実装することで、実現可能である。

【0058】

ここで、ハードウェア検証用のシミュレータが必要とするのは、信号の駆動側がいつその信号をドライブするかといったタイミングである。システム検証用のシミュレータでは、信号の駆動側ではなく、トランザクションの駆動側（以降マスター）のタイミングで、communicate を実装している。

【0059】

そこで、機能モデル 8 のインターフェースとして、トランザクションのマスターによる communicate を、図 18 のフローチャートにおける信号を駆動するインターフェースを実装した communicateMaster 処理工程 1081 と、駆動される信号に関するインターフェースを実装した communicateSlave 処理工程 1082 に分

ける。

【0060】

図11の処理工程107と処理工程108が、図12のサイクルC、サイクルC₊₁、サイクルC₊₂、サイクルC₊₃、サイクルC₊₄の各サイクル毎に呼び出される。そして、図18では、処理工程1081、処理工程107、処理工程1082が、シミュレータ制御手段2により呼び出される。処理工程108のcommunicationでは、処理工程1081のcommunication masterと処理工程1082のcommunicationslaveを連続して呼ぶか、処理工程1081と処理工程1082での機能と同様の機能を、機能モデル8が提供すればよい。

【0061】

以上、本発明の方法を用いることで、ソフトウェア検証向け、システム検証向け、ハードウェア検証向けでモデルの共通化が行える。本実施の形態では、ソフトウェア検証向け、システム検証向け、ハードウェア検証向けの抽象度として分かりやすい例を挙げたが、本発明はそれぞれの検証向けの、様々な抽象度を包含するものである。

【0062】

次に、デバッグ用インターフェースに関して述べる。

【0063】

モデルを用いたデバッグの例として、DMAブロックを対象とした場合について述べる。DMAはメモリのデータを転送する機能を実装したブロックである。すなわち、ブロックの機能としてデータ転送の開始、終了といった処理が行われた場合にブレークを行うようにするインターフェースを備えることで、デバッグ性が向上する。

【0064】

このインターフェースは、機能に対してのものである場合、各用途向けモデルで共通して利用できる。さらに、デバッグでは、ブロックの持つメモリマップドIOレジスタを含むメモリの値を参照することがある。このメモリの値の参照、変更等を行うインターフェースも、このデバッグ用インターフェースに含まれる。各検証において、バスの振舞が必要でなく、メモリの値が書き換えられれば良

い場合があり、その場合には、このインターフェースを用いて高速にシミュレーションを実施することも可能となる。

【0065】

ソフトウェア開発の場合など、更なる高速化のために、メモリ機能を、メモリを保持するブロックではなく、全てのメモリを管理するブロックに実装し、高速化を行う際には、このメモリブロックとソフトウェアが動作するブロックのみをシミュレータ制御手段から制御し、他のブロックをこの環境から分離することで、さらにシミュレーションを高速化することも可能であり、本発明の範疇に含まれる。

【0066】

また、今までの例では、シミュレータ制御手段2が一つのレベルで実装する例を述べてきたが、実際の例では複数のレベルで実装される場合もある。図19は、その一例である。シミュレータ制御手段2がシミュレータ制御手段S20とシミュレータ制御手段S21とシミュレータ制御手段S22から構成される場合を示している。

【0067】

シミュレータ制御手段S20が、シミュレータ制御手段S21とシミュレータ制御手段S22を制御し、シミュレータ制御手段S21がシミュレータモデル9とシミュレータモデル16を制御する。このとき、シミュレーションモデル9とシミュレーションモデル16を駆動するタイミングがクロックレベルである場合については、既に説明した。

【0068】

シミュレータモデル9と16で使用するクロックサイクルと、シミュレータモデル23で使用するクロックサイクルが異なる場合、制御手段S21とS22では、それぞれのクロックサイクル数ではなく、シミュレーション時間で制御を行う必要があり、その機能を実装することで、異なるクロックソースを用いるモデル混在のシミュレーション環境が実現できる。

【0069】

最後に拡張用インターフェースの例として、性能解析の例を挙げる。システム

検証では、どのバスマスタがどの程度バスを利用、占有しているか、いつどのメモリ番地にアクセスがあったか、どの程度バスの利用を待たされたかといった各種情報を利用して、性能解析を行う。このような情報を得るための拡張用インターフェースを用いて情報を得て性能解析を行い、性能解析の結果を基に再度システムをチューニングして、最適なシステムを設計することが可能となる。

【0070】

【発明の効果】

以上のように本発明によれば、大規模なシステム L S I のハードウェア検証用システムシミュレータ、ソフトウェア開発検証用シミュレータ、システム検証用システムシミュレータをはじめとする各種目的用システムシミュレータにおいて、共通のモデルが利用可能になるため、異なるシミュレータ間でモデルの共有化が図られ、統合シミュレーション制御方法によって、適切な精度のシミュレーションを同一のモデルを用いて実行できる。これによって、大規模なシステム L S I のハードウェア／ソフトウェア協調設計環境を構築するための工数を削減でき、ひいてはシステム L S I の設計期間短縮につながり、高性能、低コストのシステム L S I を提供することができる。

【図面の簡単な説明】

【図1】 本発明の実施の形態のシミュレータ装置の構成を示すブロック図

【図2】 システム L S I を示す概念図

【図3】 ハードウェア検証用シミュレータを示す概念図

【図4】 ソフトウェア検証用シミュレータを示す概念図

【図5】 システム検証用シミュレータを示す概念図

【図6】 ハードウェア検証用シミュレータで必要とされる精度を示す概念図

【図7】 ソフトウェア検証用シミュレータで必要とされる精度を示す概念図

【図8】 システム検証用シミュレータで必要とされる精度を示す概念図

【図9】 ソフトウェア検証用シミュレータ I / F を示す概念図

【図10】 従来のソフトウェア検証用シミュレータの処理工程を示すフローチャート

【図11】 システムシミュレータの処理工程を示すフローチャート

- 【図 12】 命令レベルとサイクルレベルとの違いの一部を示す説明図
- 【図 13】 命令レベルでの処理工程を示すフローチャート
- 【図 14】 命令レベルでの命令 1 の処理工程を示すフローチャート
- 【図 15】 命令レベルでの命令 2 の処理工程を示すフローチャート
- 【図 16】 命令レベルでの命令 3 の処理工程を示すフローチャート
- 【図 17】 サイクルレベルでの処理工程を示すフローチャート
- 【図 18】 ハードウェア検証用シミュレータの処理工程を示すフローチャート

ト

【図 19】 本発明の別の実施の形態のシミュレータ装置の構成を示すブロック図

【符号の説明】

- 1 検証用シミュレータ装置
- 2 シミュレータ制御手段
- 3, 10, 17 ソフトウェア検証シミュレータ用 I/F
- 4, 11, 18 ハードウェア検証用シミュレータ用 I/F
- 5, 12, 19 システム検証シミュレータ用 I/F
- 6, 13, 20 デバッグ用 I/F
- 7, 14, 21 拡張用 I/F
- 8, 15, 22 ブロックの機能モデル
- 9, 16, 23 シミュレータモデル
- 31, 32, 33, 34, 35, 36, 37 ブロック
- 60 ハードウェア検証用シミュレータ
- 61, 62, 63, 64, 65, 66, 67 ハードウェア対応の検証用モデル
- 68, 69 バスモデル
- 90, 95 ソフトウェア検証用シミュレータ
- 91, 92 ソフトウェア対応の検証用モデル
- 92, 94 仮想モデル
- 120 システム検証用シミュレータ

1 2 1, 1 2 2, 1 2 3, 1 2 4, 1 2 5, 1 2 6, 1 2 7 システム対応の
検証用モデル

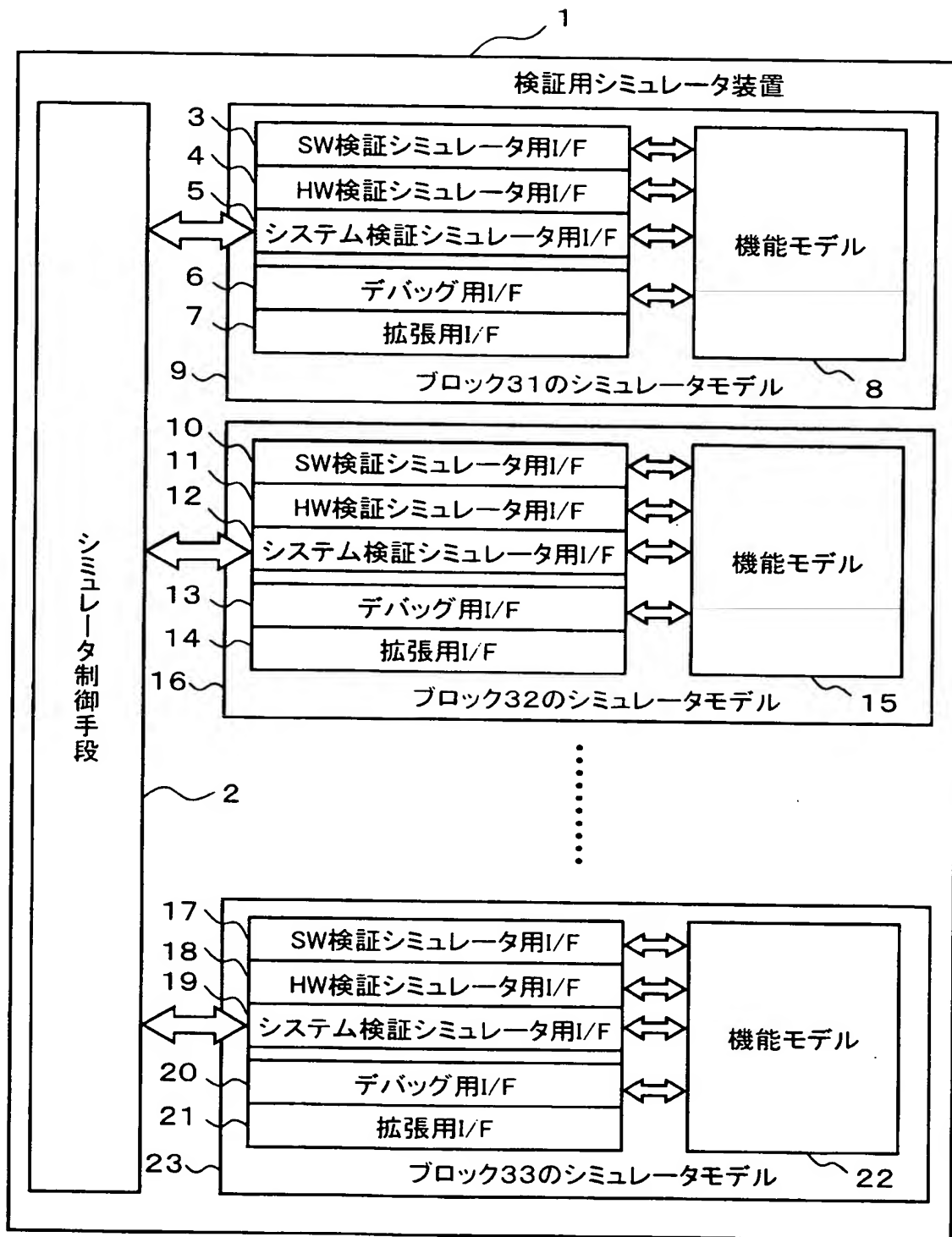
1 2 8, 1 2 9 バスモデル

S 2 1, S 2 2, S 2 3 シミュレータ制御手段

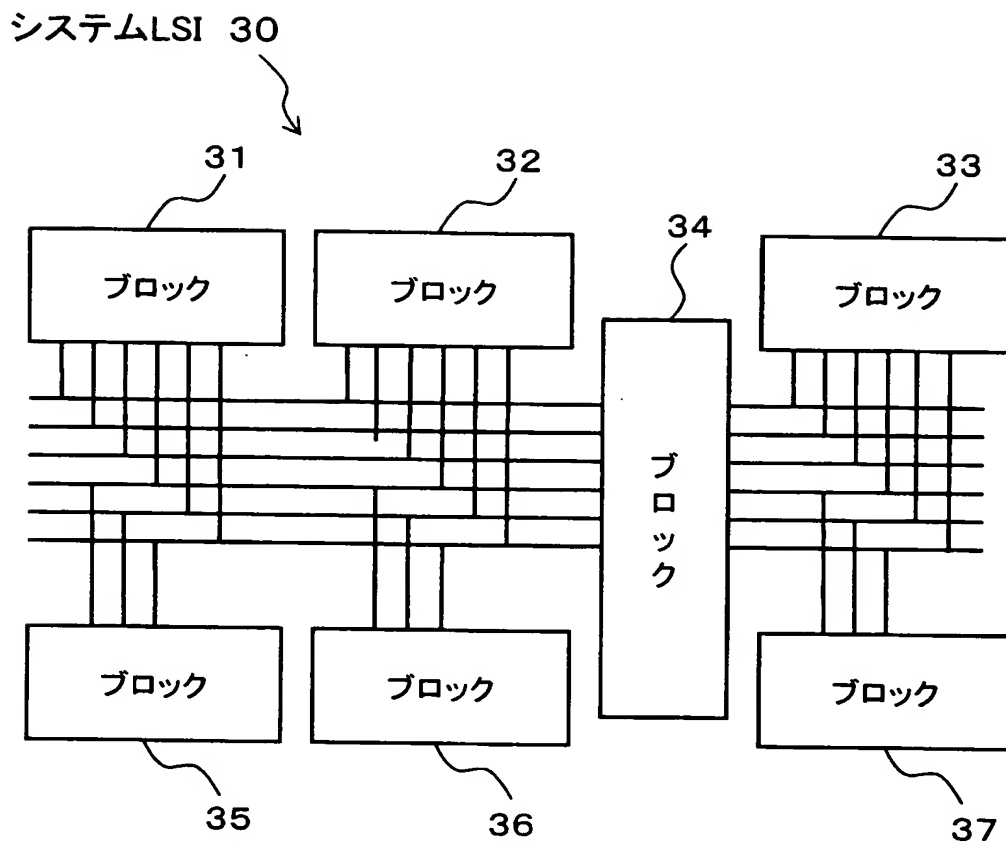
【書類名】

図面

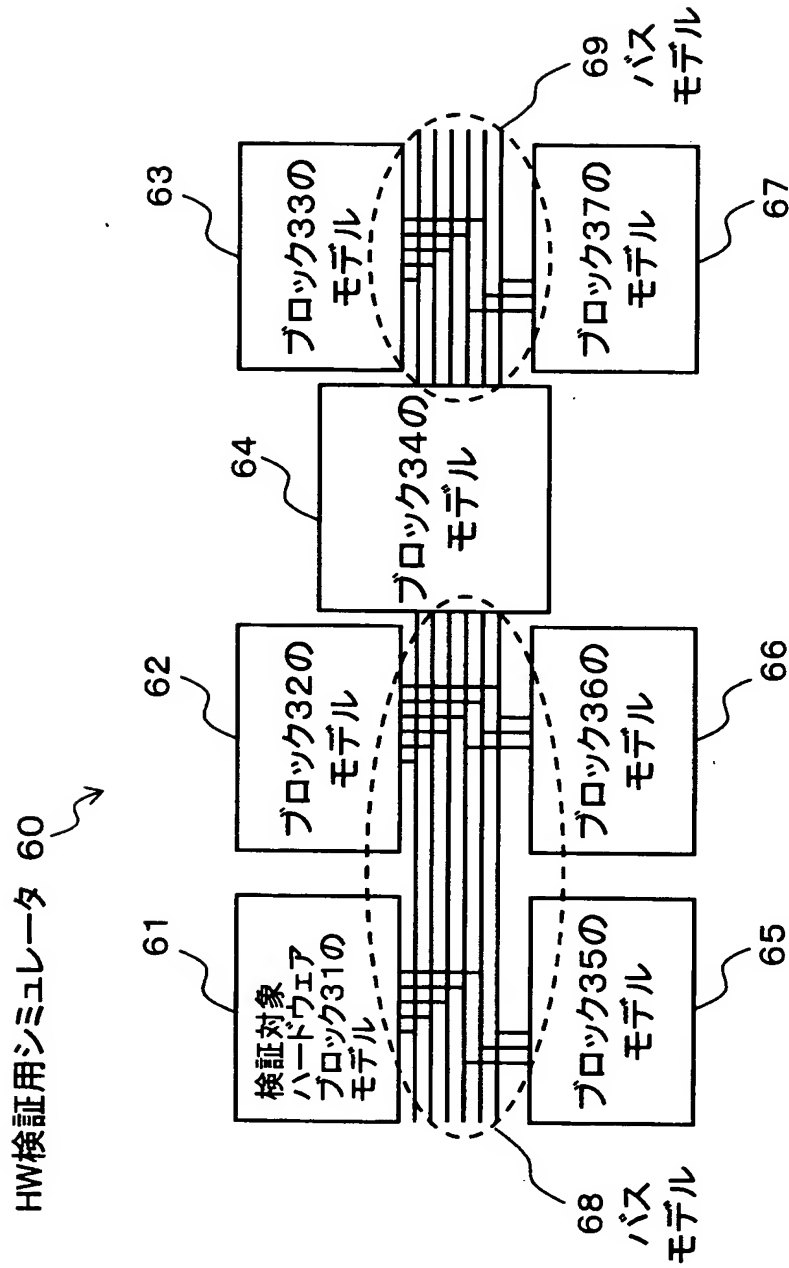
【図 1】



【図 2】

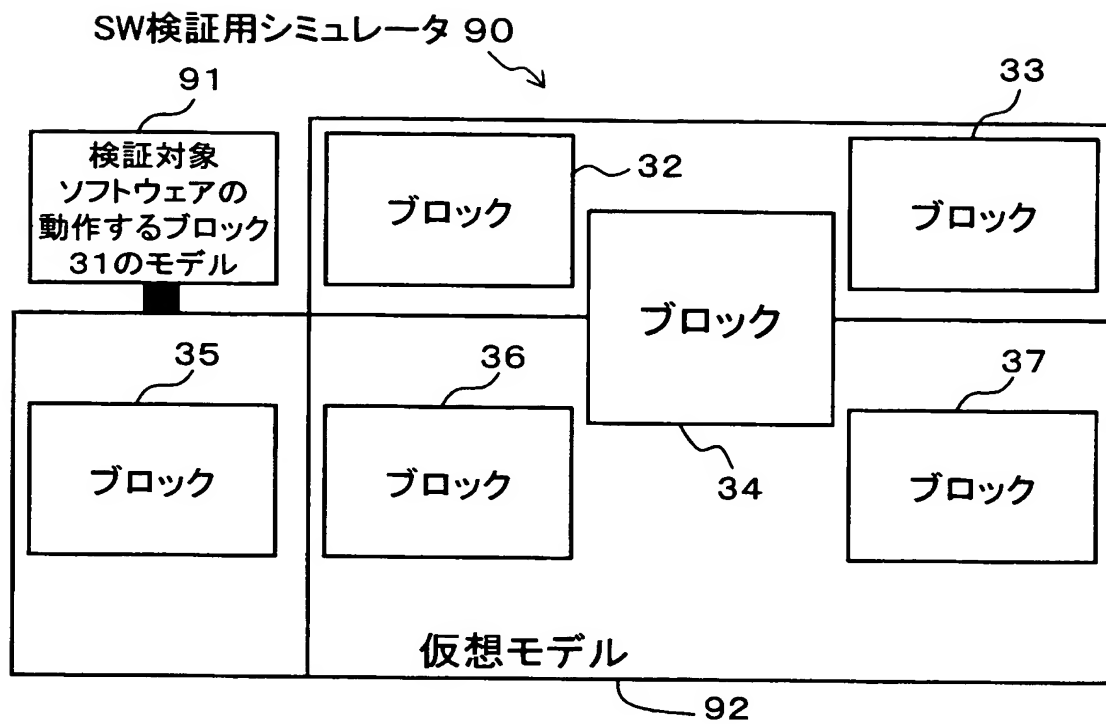


【図 3】

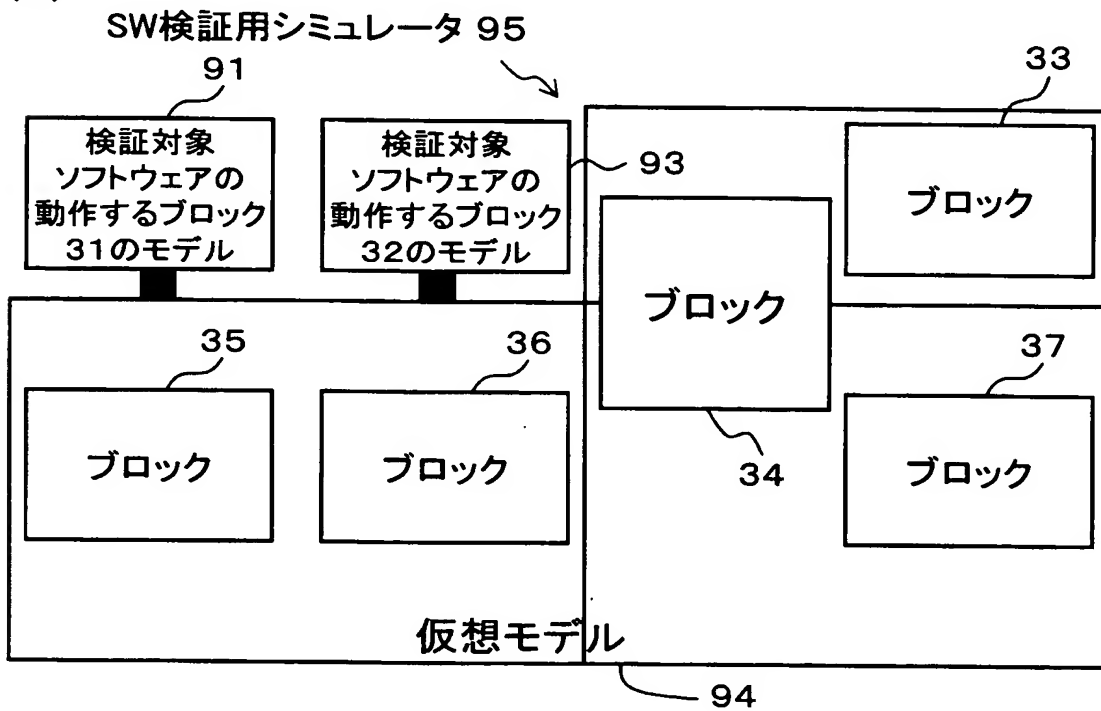


【図 4】

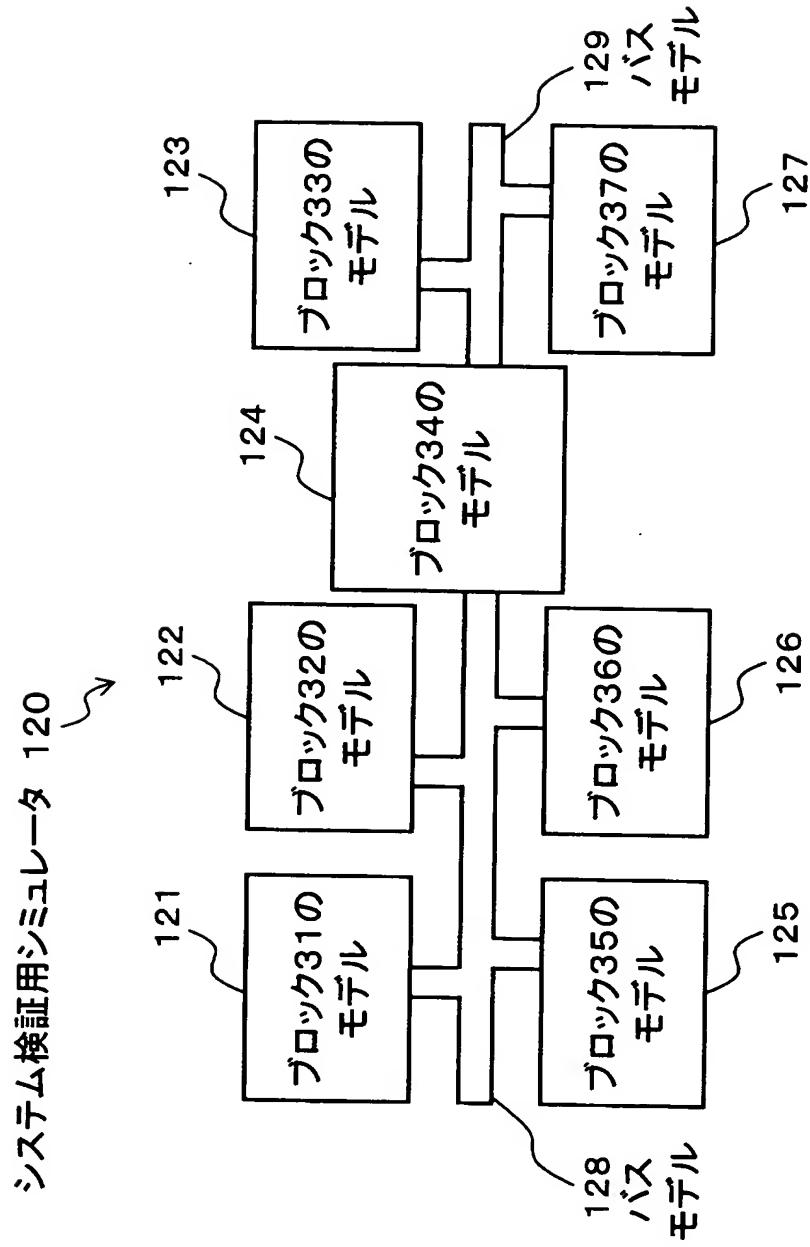
(a)



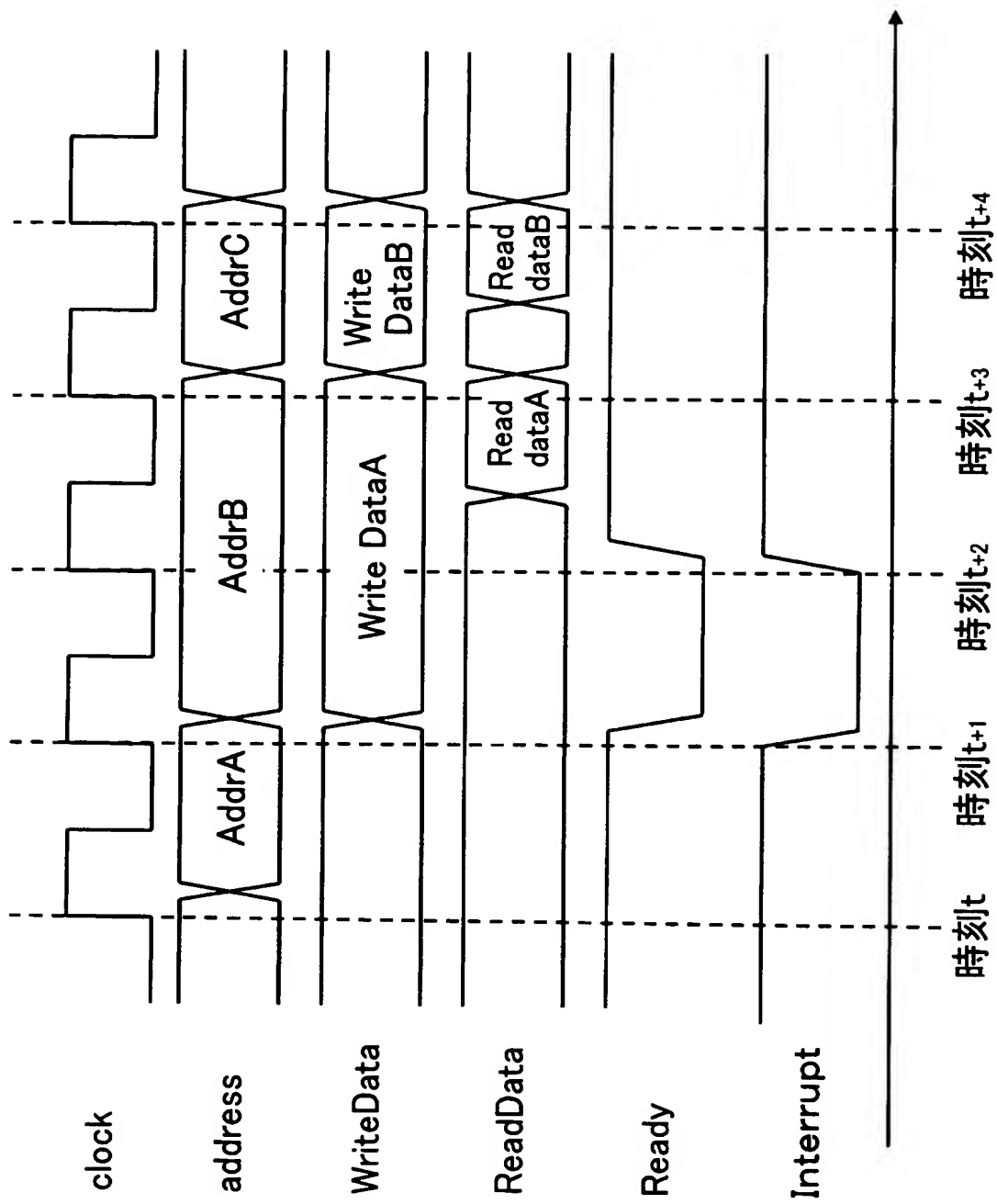
(b)



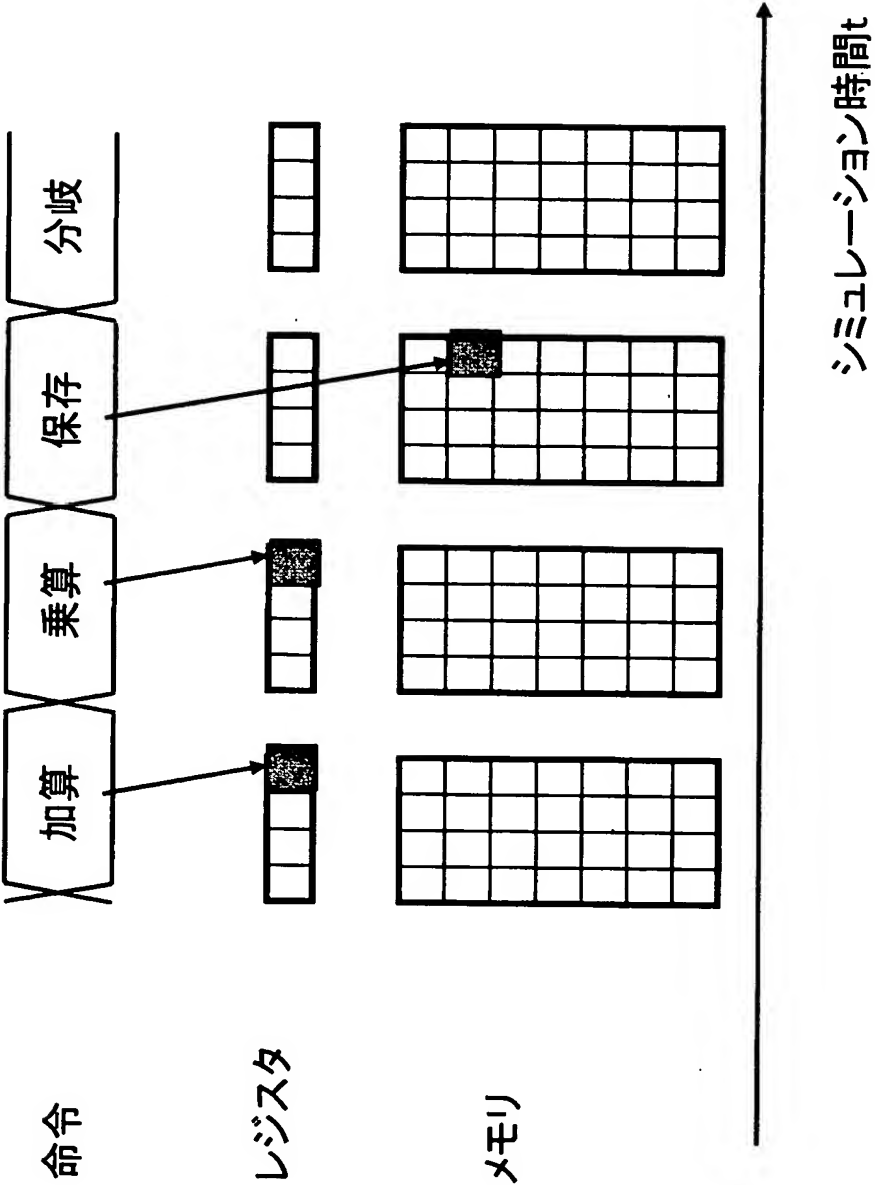
【図 5】



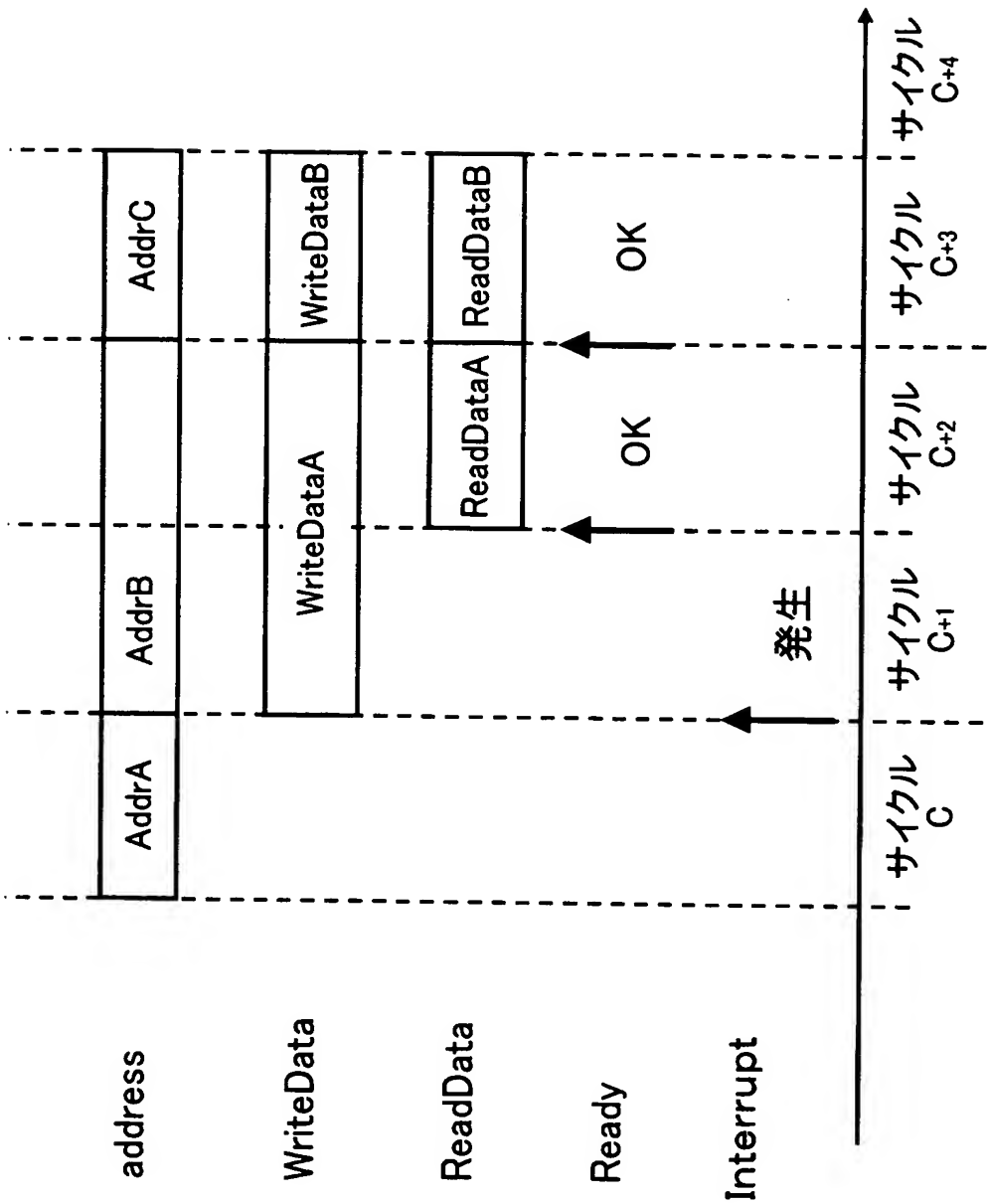
【図 6】



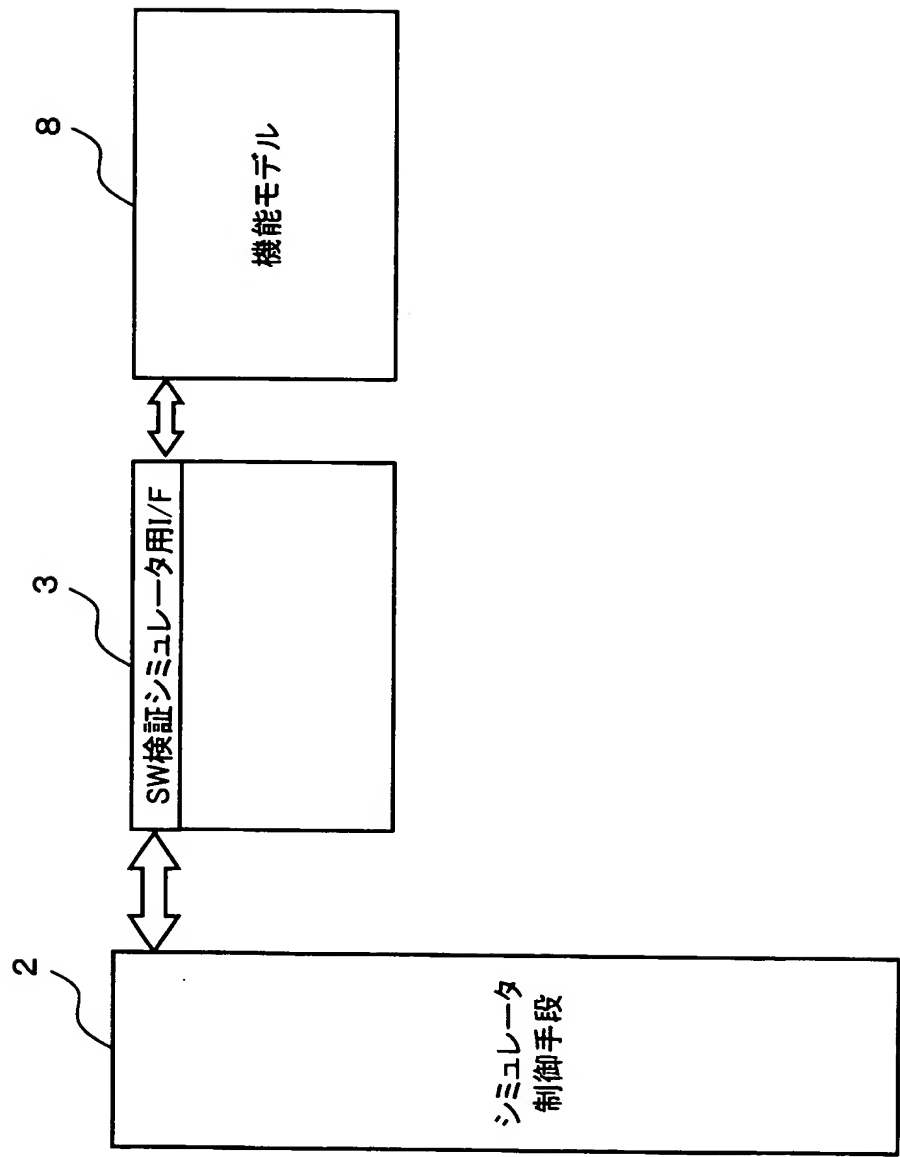
【図 7】



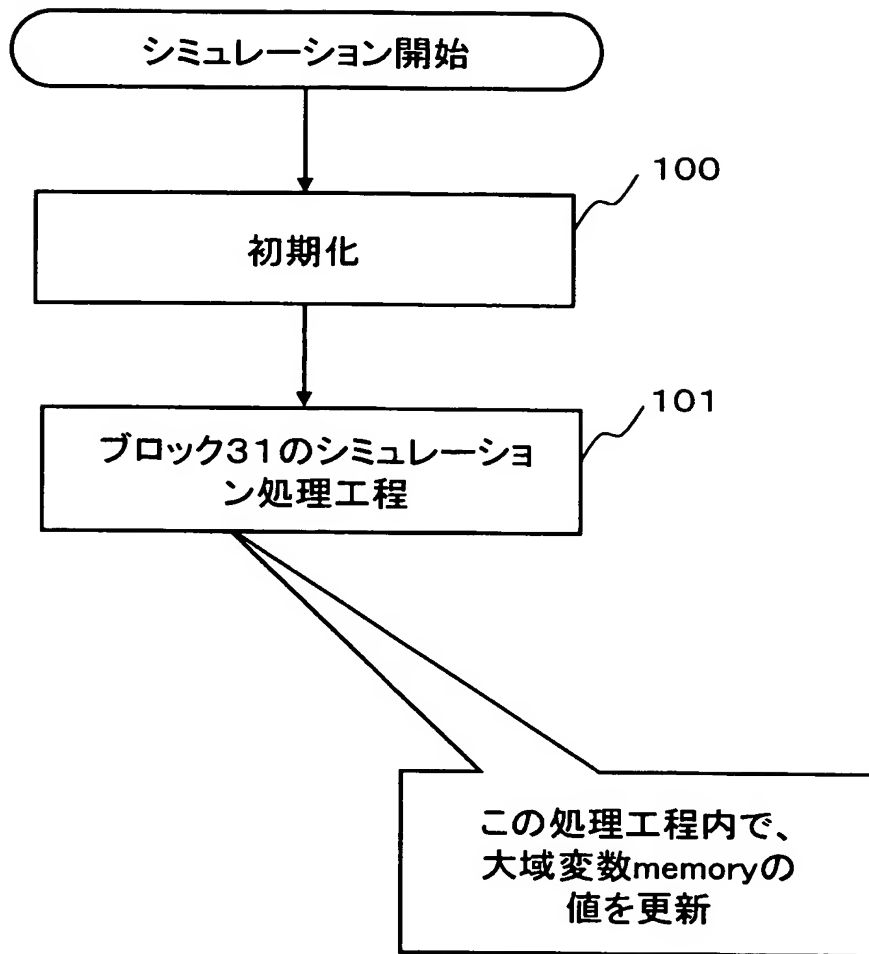
【図 8】



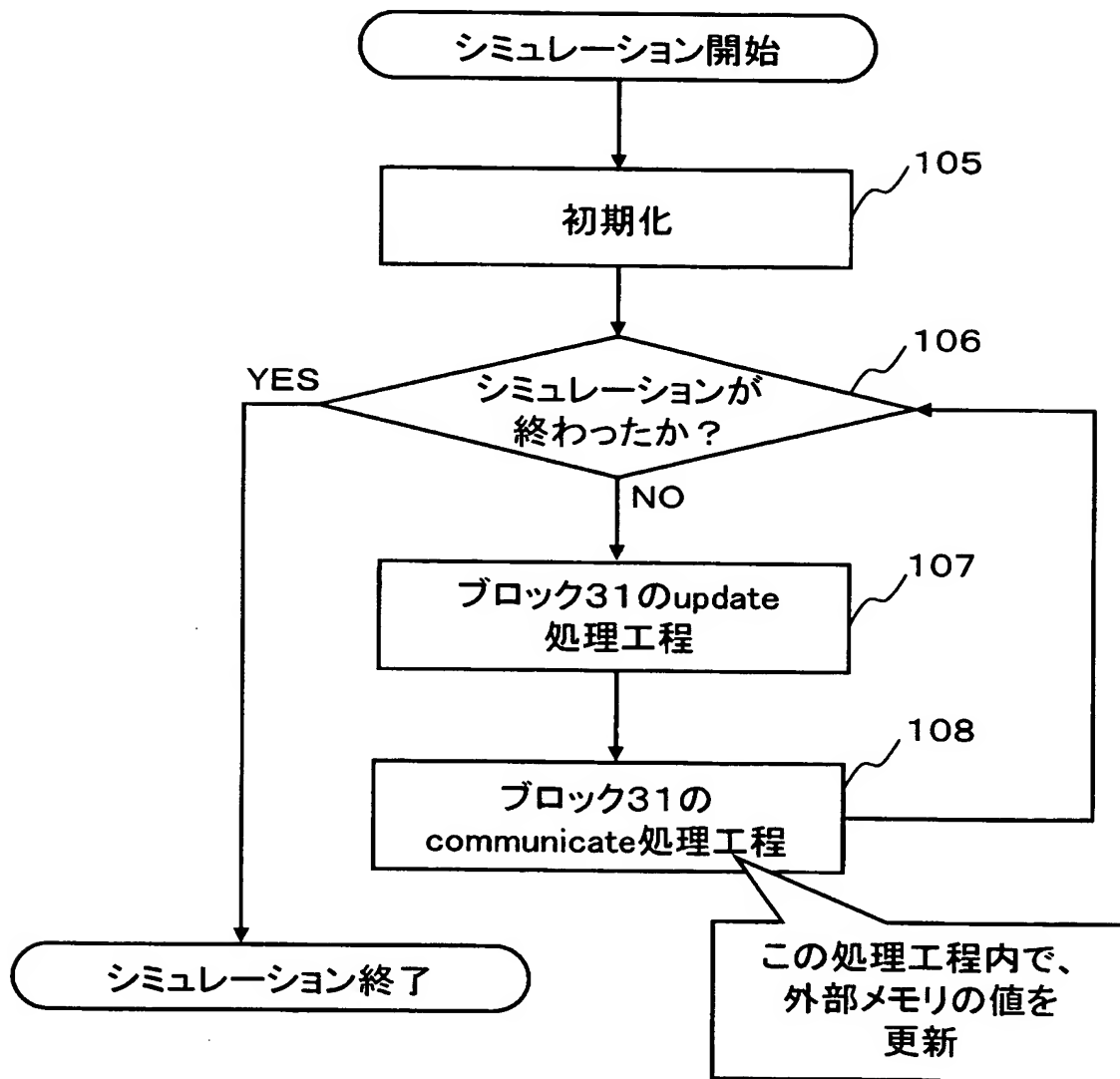
【図 9】



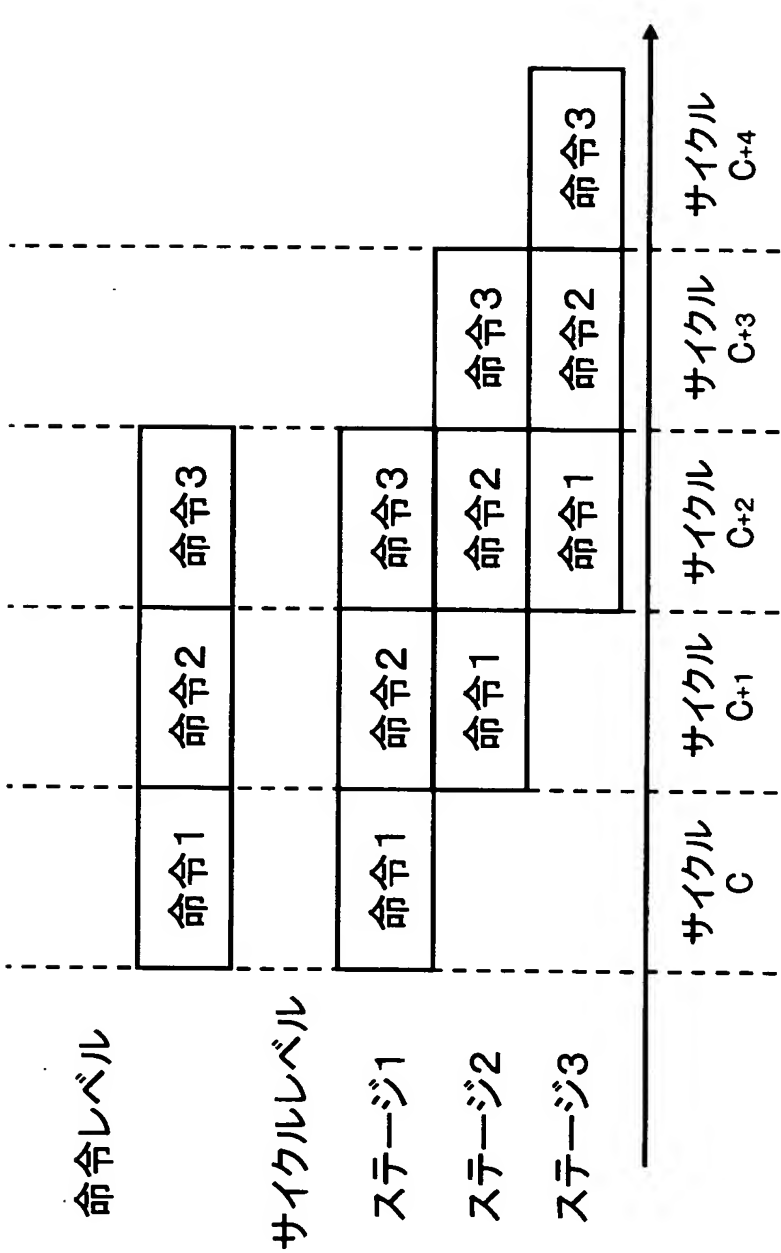
【図 10】



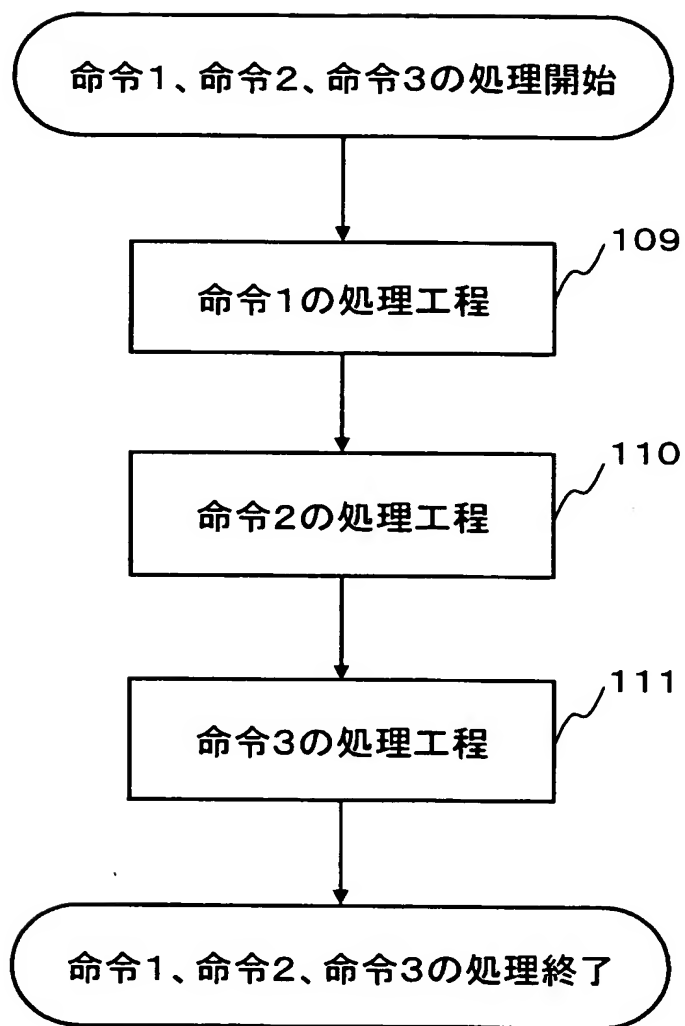
【図 11】



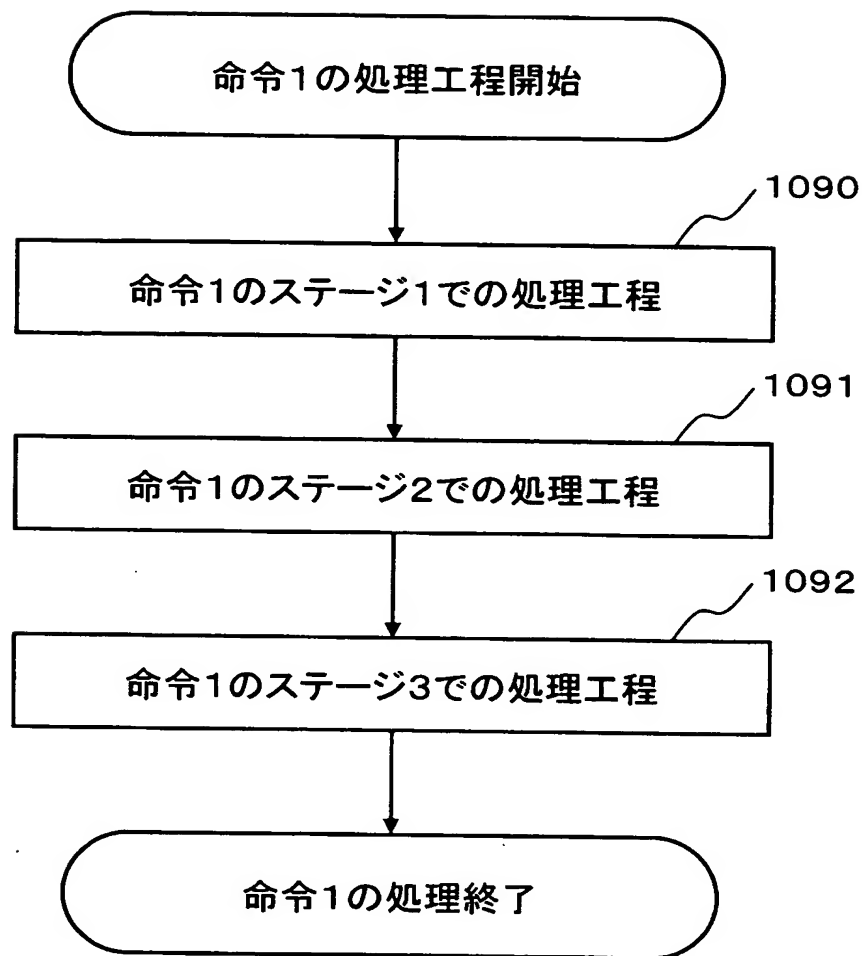
【図 12】



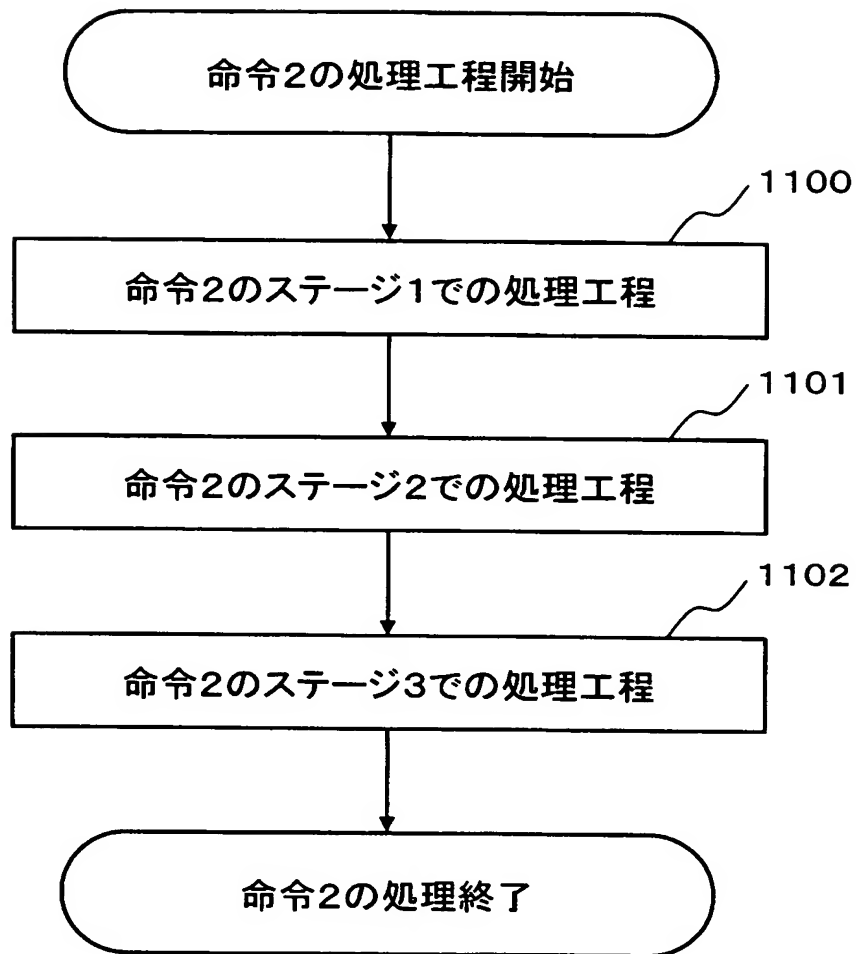
【図 13】



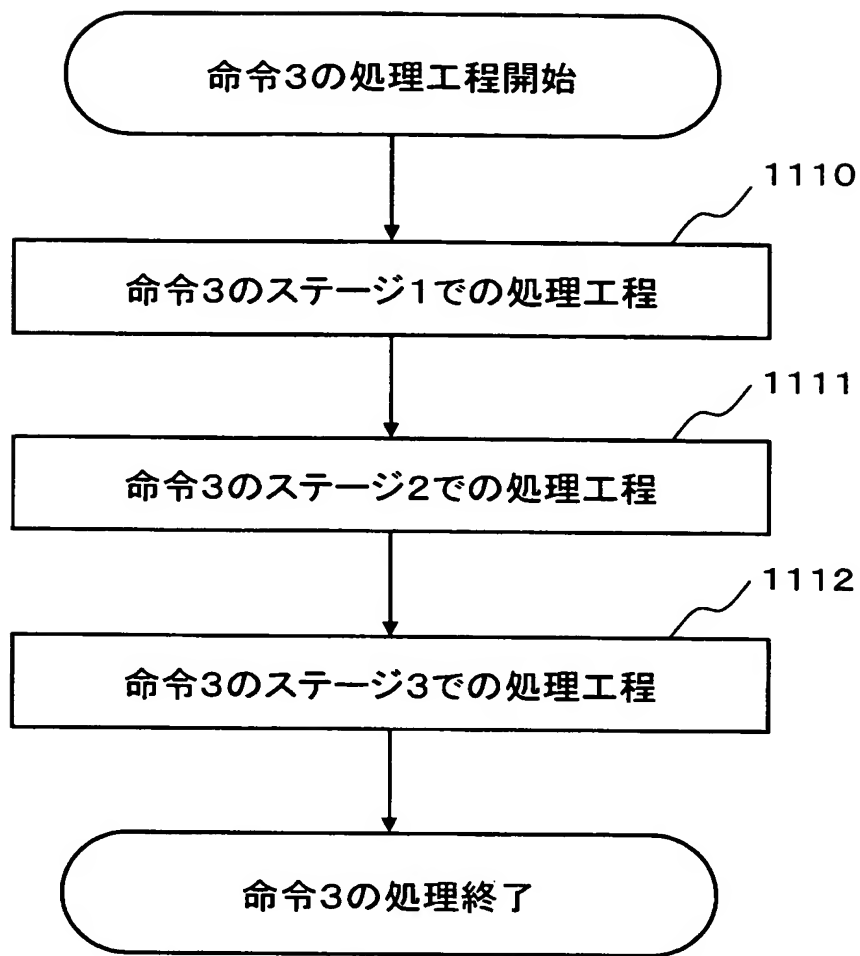
【図 14】



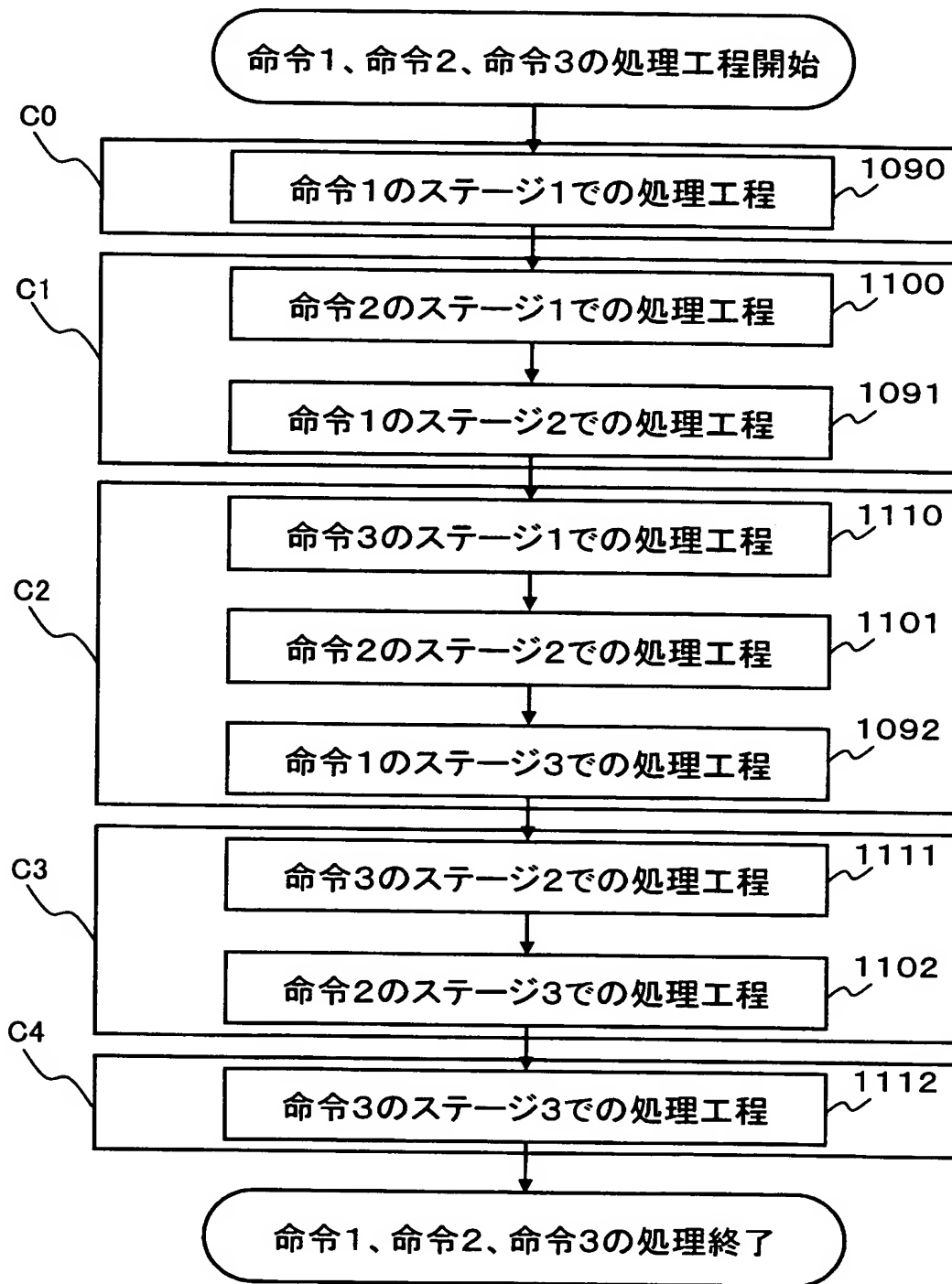
【図 15】



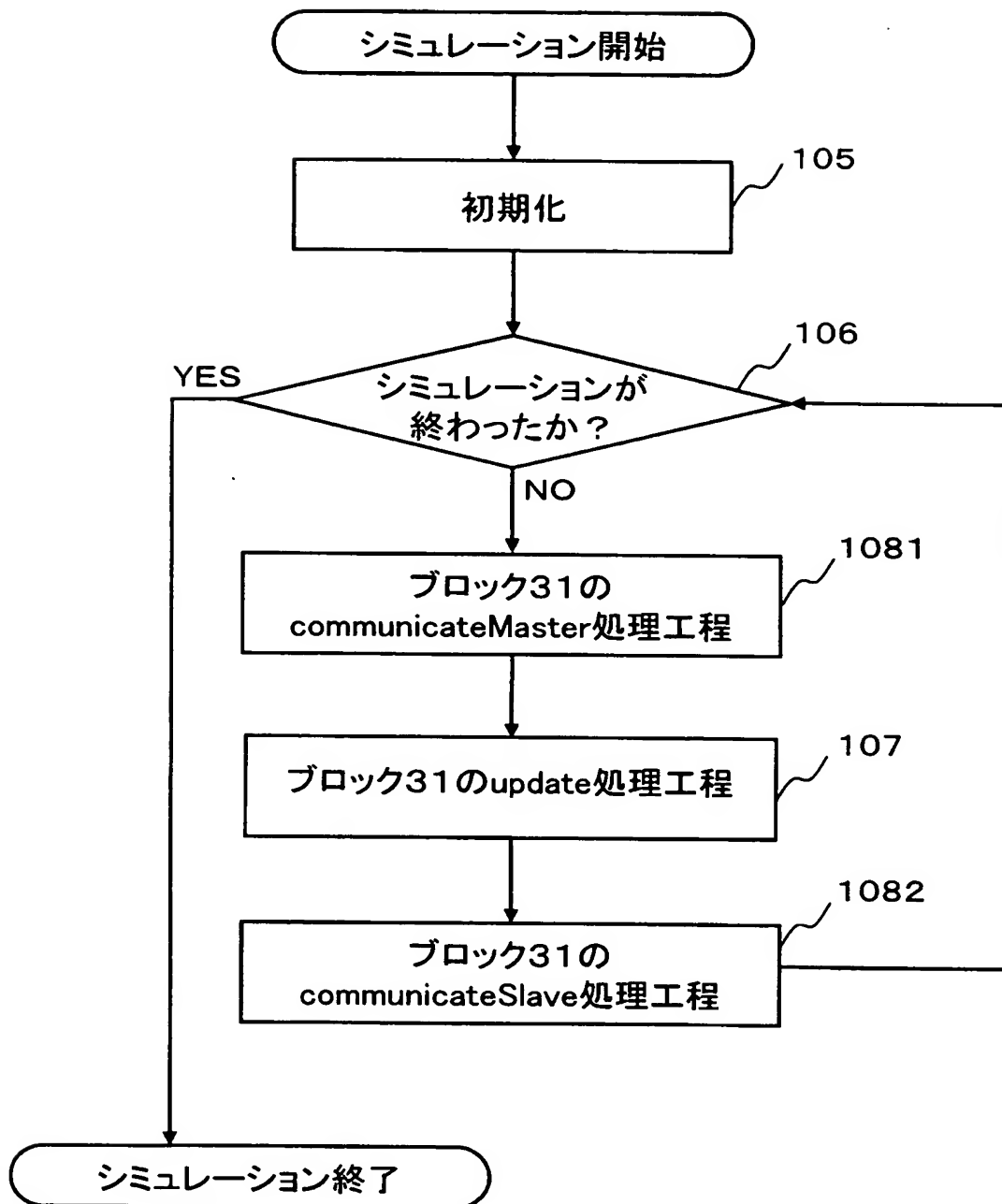
【図 16】



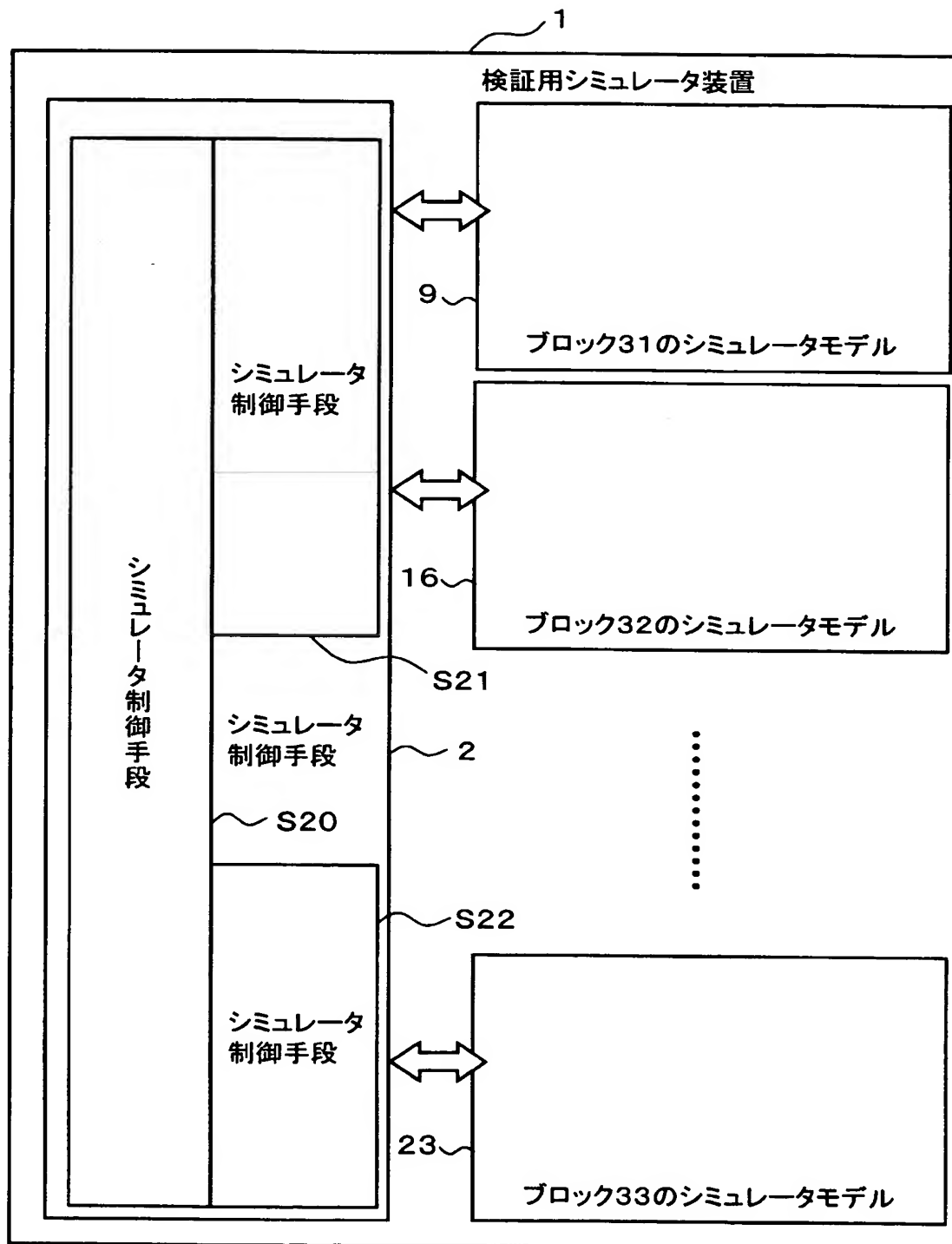
【図 17】



【図18】



【図 19】



【書類名】 要約書

【要約】

【課題】 システム L S I 開発の、それぞれの用途向けにシステム L S I を構成するブロックのモデルを用意するのが現状であるが、システム L S I の複雑化大規模化を鑑みると、それらのモデルを用意するための工数も同様に増加している。結果、ハードウェア／ソフトウェア協調設計の利点が失われてしまう。

【解決手段】 大規模なシステム L S I の 1 チップシミュレーションで用いるブロックを、異なるシミュレータの制御部から呼び出し可能なインターフェースを実装することで、シミュレータ設計の工数を削減する。また、デバッグ用のインターフェースを実装することで、各用途のための検証解析機能を実現する。

【選択図】 図 1

認 定 ・ 付 加 情 報

特許出願の番号	特願 2 0 0 3 - 1 0 4 9 0 7
受付番号	5 0 3 0 0 5 8 5 4 4 7
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 4 月 1 0 日

< 認定情報・付加情報 >

【提出日】 平成15年 4月 9日

次頁無

特願 2 0 0 3 - 1 0 4 9 0 7

出 願 人 履 歷 情 報

識別番号

[0 0 0 0 0 5 8 2 1]

1 . 変 更 年 月 日

1 9 9 0 年 8 月 2 8 日

[変 更 理 由]

新 規 登 録

住 所

大 阪 府 門 真 市 大 字 門 真 1 0 0 6 番 地

氏 名

松 下 電 器 産 業 株 式 会 社